

운영체제 03분반

실습12

201402447 한원희

```

int a[5] = {0};
int b[5] = {0};
int b_index = 0;
sem_t* isok;

void start(void);
void init(void);
void off(void);
void fail(void);
void LEDR(void);
void LEDY(void);
void LEDG(void);
void LEDW(void);
void game(void);

```

먼저 대략적으로 설명 하면 배열 a는 랜덤인 숫자가 들어가고 배열 b는 입력한 led\_r ,y, g를 숫자로 배열 b에 넣어주어서 나중에 a와 b를 비교하여 입력한 결과가 맞았는지 틀렸는지를 봐 주는 역할을 합니다.

```

int main(void){
    init();
    start();

    int status;
    pthread_t pt[4];

    sem_unlink("isok");
    if((isok = sem_open("isok", O_CREAT, 0644, 0)) == SEM_FAILED){
        perror("Sem failed");
        exit(1);
    }

    pthread_create(&pt[0], NULL, (void*)LEDR, NULL);
    pthread_create(&pt[1], NULL, (void*)LEDY, NULL);
    pthread_create(&pt[2], NULL, (void*)LEDG, NULL);
    pthread_create(&pt[3], NULL, (void*)LEDW, NULL);

    game();

    return 0;
}

```

main입니다. 각 입력 버튼마다 pthread가 생성되고 semaphore를 open해 줍니다. 그 이후 game이 실행 됩니다.

```

void start(void){
    int i = 0;

    while(i < 2){
        digitalWrite(LED_R, 1);
        delay(250);
        digitalWrite(LED_R, 0);

        digitalWrite(LED_Y, 1);
        delay(250);
        digitalWrite(LED_Y, 0);

        digitalWrite(LED_G, 1);
        delay(250);
        digitalWrite(LED_G, 0);
        i++;
    }
}

```

start메소드는 게임이 시작할 때, 성공했을 때 실행해주는 불빛을 순서대로 실행해주는 메소드 입니다.

```

void init(void){
    if(wiringPiSetup() == -1){
        puts("Setup Fail");
        exit(1);
    }
    pinMode(SW_R, INPUT);
    pinMode(SW_Y, INPUT);
    pinMode(SW_G, INPUT);
    pinMode(SW_W, INPUT);
    pinMode(LED_R, OUTPUT);
    pinMode(LED_Y, OUTPUT);
    pinMode(LED_G, OUTPUT);
    off();
}

```

각 gpio 핀에 대해 입출력 모드를 설정해 주는 메소드입니다.

```

void off(void){
    digitalWrite(LED_R, 0);
    digitalWrite(LED_Y, 0);
    digitalWrite(LED_G, 0);
}

void fail(void){
    int i = 0;

    while(i < 3){
        delay(150);
        digitalWrite(LED_R, 1);
        digitalWrite(LED_Y, 1);
        digitalWrite(LED_G, 1);
        delay(150);
        off();

        i++;
    }
}

```

led불을 모두 꺼주는 off 메소드와 게임을 실패했을 때 동시에 불을 켜주고 끄는 메소드 입니다.

```

void game(void){
    sem_t ok;

    srand(time(NULL));
    int g = 0;
    int h;

    while(g < 5){
        h = g + 1;

        for(int i = 0; i < h; i++){
            a[i] = rand() % 3 + 1;
        }
    }
}

```

게임을 처음 시작하면 제일 먼저 while문을 통해 round를 시작해 줍니다. H의 값은 round를 처음 시작하면 led가 1개 들어오고 round2 일때는 2개 씩 들어오니까 g + 1을 해줍니다. Srand와 rand % 3 + 1을 통해 항상 랜덤 인 1 ~ 3값을 배열 a에 순서대로 넣어줍니다. 놓여지지 않는 값에는 0으로

채워져 있습니다.

```
for(int i = 0; i < h; i++){
    if(a[i] == 1){
        delay(500);
        digitalWrite(LED_R, 1);
        delay(500);
        digitalWrite(LED_R, 0);

    }
    else if(a[i] == 2){
        delay(500);
        digitalWrite(LED_Y, 1);
        delay(500);
        digitalWrite(LED_Y, 0);

    }
    else if(a[i] == 3){
        delay(500);
        digitalWrite(LED_G, 1);
        delay(500);
        digitalWrite(LED_G, 0);

    }
    else
        break;

}
```

그 이후 배열 a에 랜덤으로 넣어진 값을 led를 통해서 알려주는 파트입니다. 값이 1이라면 led\_r, 2라면 led\_y, 3이라면 led\_g입니다.

```

sem_wait(isok);

for(int i =0; i < 5; i++){
    if(a[i] == b[i])
        continue;
    else{
        fail();
        exit(1);
    }
}

for(int i = 0; i < 5; i++){
    b[i] = 0;
}

b_index = 0;

g++;
}
start();
exit(1);

```

그 이후 값을 받아와 검사해주는 파트입니다. 흰색 버튼이 눌릴때까지 검사를 시작하면 안되므로 sem\_wait을 통해서 흰색 버튼이 눌리는 것을 기다려 줍니다. 그이후 첫번째 for문에서 랜덤으로 받아온 a배열과 입력으로 받은 b배열을 검사해주고 맞다면 계속해서 실행해주고 아니라면 fail과 exit을 실행해 줍니다. 검사가 끝났다면 다음 round를 위해 b배열을 0으로 다시 바꿔주고 b\_index 또한 0으로 바꿔줍니다.

만약 round가 모두 끝나 while문을 빠져 나왔다면 게임을 클리어 한 것이므로 start를 실행해주고 종료해 줍니다.

```

void LEDR(void){
    while(1){
        if(digitalRead(SW_R) == 0){
            digitalWrite(LED_R, 1);
            delay(250);
            digitalWrite(LED_R, 0);
            b[b_index] = 1;
            b_index += 1;

        }

        sleep(1);
    }
}

```

버튼을 통해 입력받은 값을 배열b에 넣어주는 메소드입니다. 버튼이 눌렀다면 b배열에 led\_r이라면 1을, y라면 2를, g라면 3을 넣고 b\_index를 하나 증가시켜 줍니다. 위의 메소드는 r,y,g 가 생긴 것이 비슷합니다.

```

void LEDW(void){

    while(1){
        if(digitalRead(SW_W) == 0){
            sem_post(&isok);
            delay(500);
        }
        sleep(1);
    }

}

```

마지막으로 하얀색 버튼이 눌렀을 때 동작입니다. 하얀색 버튼이 눌렀다는 의미는 b배열에 입력값을 넣는 동작을 마쳤다는 의미로 sem\_post를 하여 다시 게임으로 돌아가 a배열과 b배열을 검사하는 과정으로 돌아가는 역할을 수행합니다.